

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-254890

(43)Date of publication of application : 25.09.1998

(51)Int.Cl.

G06F 17/30

G06F 13/00

G06F 15/00

(21)Application number : 09-055716

(71)Applicant : FUJITSU LTD

(22)Date of filing : 11.03.1997

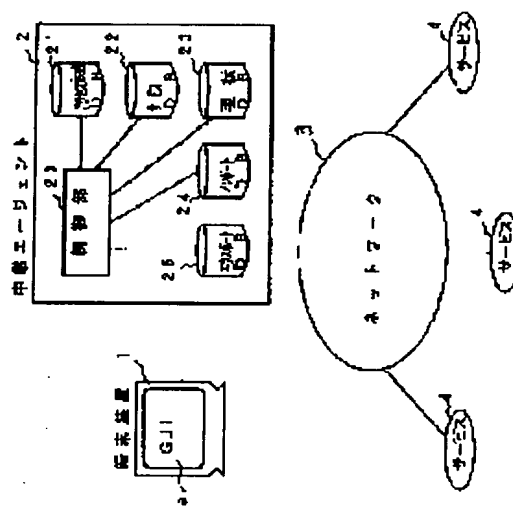
(72)Inventor :
WADA YUJI
USHIJIMA SATORU
ICHIKI HIROMOTO
SHIOUCHI MASATOSHI
TAKADA HIROSHI
MORI TAKAO

(54) ACCESS SYSTEM TO VARIOUS KINDS OF SERVICES USING NETWORK

(57)Abstract:

PROBLEM TO BE SOLVED: To enabling starting and finishing a service and updating service contents according to need by simply accessing to a necessary service in an access system to various services utilizing a network to which a plurality of computers respectively providing service are connected.

SOLUTION: The system is provided with a repeating agent 2 connected to the network and receiving all access to the services of the computers from a terminal device. The agent 2 is provided with an accessing method data base 21 where the name of a service, the name of a computer provided with the service and its accessing method are registered, retrieves the data base with respect to an access request designating a service from the terminal device, establishes connection between with the computer to provide the service based on the registered accessing method and starts a program providing the service.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-254890

(43) 公開日 平成10年(1998) 9月25日

(51) Int.Cl.⁸

識別記号

F I

G 0 6 F 17/30

13/00

15/00

3 5 5

3 1 0

G 0 6 F 15/40

13/00

15/00

15/40

3 1 0 F

3 5 5

3 1 0 D

3 8 0 E

審査請求 未請求 請求項の数 7 O L (全 21 頁)

(21) 出願番号

特願平9-55716

(22) 出願日

平成9年(1997) 3月11日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72) 発明者 和田 裕二

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72) 発明者 牛嶋 悟

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74) 代理人 弁理士 穂坂 和雄 (外2名)

最終頁に続く

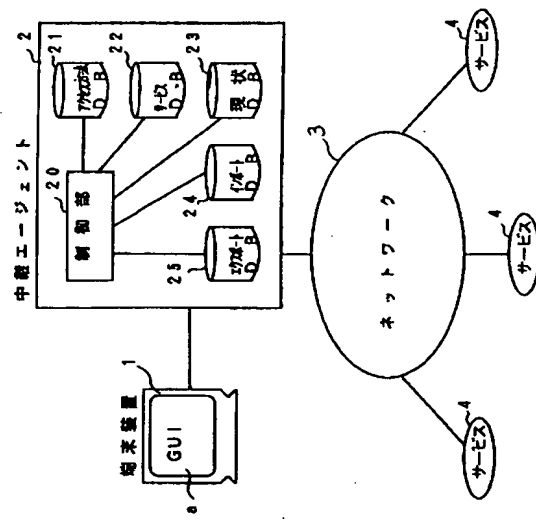
(54) 【発明の名称】 ネットワークを利用した各種サービスへのアクセス方式

(57) 【要約】

【課題】 本発明はそれぞれサービスを提供する複数の計算機が接続されたネットワークを利用した各種サービスへのアクセス方式に関し、必要なサービスに対して簡単にアクセスすることができ、必要に応じたサービスの起動、終了及びサービス内容の更新を行うことができることを目的とする。

【解決手段】 ネットワークに接続されると共に端末装置からの前記計算機のサービスへのアクセスを全て受け付ける中継エージェントを設ける。中継エージェントは、サービス名とサービスが提供される計算機名とそのアクセス方法が登録されたアクセス方法データベースを備え、端末装置からのサービスを指定したアクセス要求に対しアクセス方法データベースを検索して、登録されたアクセス方法を元にサービスを提供する計算機との接続を確立し、サービスを提供するプログラムを起動するよう構成する。

本発明の原理構成図



【特許請求の範囲】

【請求項 1】 それぞれサービスを提供する複数の計算機が接続されたネットワークを利用した各種サービスへのアクセス方式において、前記ネットワークに接続されると共に端末装置からの前記計算機へのサービスへのアクセスを全て受けつける中継エージェントを設け、前記中継エージェントは、サービス名とサービスが提供される計算機名とそのアクセス方法が登録されたアクセス方法データベースを備え、端末装置または他中継エージェントからのサービスを指定したアクセス要求に対し、前記アクセス方法データベースを検索して、登録されたアクセス方法を元にサービスを提供する計算機との接続を確立し、サービスを提供するプログラムを起動してサービスを提供することを特徴とするネットワークを利用した各種サービスへのアクセス方式。

【請求項 2】 請求項 1 において、前記中継エージェントは、サービスを提供する各プログラムをサービス名、プログラムまたはモジュール及びプログラムタイプのデータにより登録されたサービスデータベースを備え、端末装置または他中継エージェントから要求されたサービスが前記サービスデータベースに登録されていることを検出すると、当該プログラムが既に起動していればそのサービスを利用し、起動してないと当該プログラムを起動してサービスを提供することを特徴とするネットワークを利用した各種サービスへのアクセス方式。

【請求項 3】 請求項 2 において、前記中継エージェントは、前記サービスデータベースの検索により、要求されたサービスを提供するプログラムがネットワーク上の他の計算機に存在することが判別すると、前記他の計算機から前記プログラムの複製を受け取ってサービス名とプログラムとを登録されるインポートデータベースと、他の計算機にサービスを提供するプログラムの複製を渡した時に、他の計算機名とサービス名を登録するエクスポートデータベースを備え、他の計算機から受け取った中継エージェントは、前記インポートデータベースに登録されたプログラムを起動することにより要求されたサービスを提供することを特徴とするネットワークを利用した各種サービスへのアクセス方式。

【請求項 4】 請求項 1 ～ 3 の何れかにおいて、前記中継エージェントは、サービス名、プログラム及びプログラムのタイプが指定されたプログラムを起動する時、タイプ対応の起動手順が格納された起動方法テーブルを用いて起動することを特徴とするネットワークを利用した各種サービスへのアクセス方式。

【請求項 5】 請求項 3 において、前記中継エージェントは、サービス名を指定したサービスの更新要求に対し、サービスデータベースに登録された該当サービスのデータを検索して、該当サービスがあるとこれを削除して、更新後のデータを登録すると共に、前記エクスポートデータベースに前記サービス名が登録されているか検

索し、該当サービス名があると登録された他の計算機のインポート DB の該当サービスを削除して更新後のデータを登録することを特徴とするネットワークを利用した各種サービスへのアクセス方式。

【請求項 6】 請求項 1 ～ 5 の何れかにおいて、前記中継エージェントは、現在起動しているサービスを提供するプログラムが登録された現状データベースを備え、サービス状況の検索、登録、削除の要求に応じて、前記現状データベースを検索、登録及び削除を行うことを特徴とするネットワークを利用した各種サービスへのアクセス方式。

【請求項 7】 請求項 1 ～ 6 の何れかにおいて、前記中継エージェントは、予め登録された個人の利用者が登録される利用者チェック部を備え、前記チェック部は、端末装置からのアクセス時に要求依頼者名と要求内容とを含む情報を受け取ると、要求依頼者名が登録された利用者名と一致することを検出すると前記要求内容を受け付けることを特徴とするネットワークを利用した各種サービスへのアクセス方式。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はネットワークを利用した各種サービスへのアクセス方式に関し、全世界に広がるような広域ネットワークによって複数の計算ノードが繋がっていて、各計算ノード上で様々なサービスが提供される環境における各種サービスへのアクセス方式に関する。

【0002】近年、計算機の技術の発展に伴い、ネットワークで接続された複数の計算機上で様々なサービスが提供されるようになって来ている。例えば、ネットワーク上でスケジュールを管理するシステム等は、スケジュール管理のサービスを提供していると考えることができる。このようにネットワークで計算機が接続され、それらの計算機上で種々のサービスが提供された環境において、サービスに対する簡単なアクセスや、必要に応じたサービスの起動、終了、更新等ができることが望まれている。

【0003】本発明が対象とする環境は、ネットワークによって複数の計算ノードが相互接続されているような環境である。このようなネットワークには、従来、インターネットのような全世界に広がるコンピュータ・ネットワークや LAN (Local Area Network) のような小規模なネットワーク等がある。この場合の計算ノードとしては、ワークステーション、パーソナル・コンピュータ、携帯端末等のアプリケーションを実行する計算主体を表し、以下、これらを総称して計算機と呼ぶ。

【0004】

【従来の技術】通常ネットワーク上のサービスがそれぞれ独自のアクセス手順（プロトコル）を持つ場合には、それぞれのサービスに対応して決まった手順に従わなけ

10

20

30

40

50

ればならない。

【0005】他方、各サービスがある程度統一的な手順を決め、その手順に従えば多くのサービスを利用可能にする方法がある。その例として、インターネットのWWW (World Wide Web) のブラウザ等の場合、URL (Uniform Resource Locator) と呼ばれるアドレスを用いる方法でサービスを指定してアクセスすることが可能である。指定されたサービス側では、HTTP (Hyper Text Transfer Protocol) と呼ばれるプロトコルに従うこと

によって利用者に対してサービスを提供することが可能である。

【0006】

【発明が解決しようとする課題】上記の従来の技術を用いてサービスを提供する計算機にアクセスする場合、次のような問題がある。

【0007】 サービスに対するアクセスの問題

サービスを提供する計算機側が、上記のHTTPのプロトコルに従ってサービスを提供していない場合には、この方法ではサービスを受けることが不可能である。また、たとえ、そのプロトコルでサービスを提供している場合でも、URLで指定された側が何らかの理由によりそのサービスを提供可能でない状態(サービスを提供するプログラムが起動されてない、サービスを提供するプログラムを起動している計算機にアクセスできない等)の時には、サービスを受けることが不可能である。そのような場合でも、利用者がアクセス可能な計算機のキャッシュメモリにサービスを提供する計算機から取り出したプログラム等が保持されている時には、当該他の計算機からサービスを提供することが可能であるが、キャッシュメモリにない情報にアクセスする場合や、情報が利用者との相互作用により動的に生成される場合等はキャッシュメモリを用いることが不可能である。

【0008】 必要に応じたサービスの起動、終了、更新の問題

従来の方法ではサービスを提供する側がそのサービスを提供可能な状態でない場合は、そのサービスを提供するプログラム等を起動しない限りそのサービスにアクセスすることは困難である。しかし、例えば、そのサービスがどのような計算機でどのようなプログラムによって起動されているのか等の情報が無い状況では、そのサービスを提供するプログラムを起動することは不可能である。同様に、サービスを終了させるために起動されているプログラムを終了させたり、サービスの内容を更新するために起動されているプログラムを更新することも不可能である。

【0009】本発明は、ネットワークで繋がった複数の計算機上で様々なサービスが提供される環境において、必要なサービスに対して簡単にアクセスすることができ、必要に応じたサービスの起動、終了及びサービス内容の更新を行うことができるネットワークを利用した各

種サービスへのアクセス方式を提供することを目的とする。

【0010】

【課題を解決するための手段】本発明はネットワークで接続された複数の計算機上で提供される種々のサービスに対してアクセスする方法を管理するためのサーバとして機能する機構(中継エージェントと呼ぶ)を用い、その中継エージェントを経由してサービスにアクセスすることを原理とする。

【0011】図1は本発明の原理構成を示す図である。図中、1は利用者の端末装置(ワークステーション、パーソナル・コンピュータ、携帯端末等)、1aは端末装置1上で動作し中継エージェント2との間のインタフェースとして機能するGUI(グラフィカル・ユーザ・インタフェース、例えばWWWブラウザ)、2は利用者の端末装置1や、他の計算機(ホスト)とネットワークにより接続された中継エージェント、20は中継エージェントの制御部、21は各サービスに対応するアクセス方法が登録されたアクセス方法データベース(DBで表す、以下同じ)、22はサービスを提供するプログラムが登録されたサービスデータベース(DB)、23は利用者の要求によって現在起動しているサービスを提供するプログラムが登録された現状データベース(DB)、24はサービスデータベース22に登録されている情報を元に他の計算機上に起動すべきプログラムがあることが分かった場合に、他の計算機からコピーしてきた起動すべきプログラムが登録されたインポートデータベース(DB)、25は他の計算機に起動するプログラムのコピーを渡したことを登録したエクスポートデータベース(DB)、3はネットワーク、4はネットワーク上の複数の計算機で提供される各種のサービスである。

【0012】利用者がネットワーク3上の各種サービス4の一つにアクセスする場合、サービスアクセスに関する要求は全て中継エージェント2に送られる、中継エージェント2は制御部20の制御によりサービス名をキーとしてアクセス方法DB21を検索し、登録されているアクセス方法を元に適切な接続を確立する。また、中継エージェント2はサービスを提供するプログラム自体をサービスDB22を用いて管理しており、例えばスケジュール管理のサービスを提供するプログラムをサービスDB22に登録しておけば、既にスケジュール管理サービスが利用可能な場合にはそのサービスを利用し、そうでない場合にはサービスDB22に登録されているプログラムを新たに起動する。但し、サービスDB22が管理する対象は、プログラムとは限らず、モジュールのようなものでもよい。現状DB23には、現在利用者の要求によって起動したサービスを提供するプログラムが登録されている。

【0013】インポートDB24はサービスDB22に登録されている情報を元に他の計算機上に起動すべきプ

プログラムがあることが分かった場合に、他の計算機からプログラムのコピーをしてきた起動すべきプログラムが登録されている。エクスポートDB25は他の計算機に対して起動するプログラムのコピーを渡したことを登録してある。これらの、インポートDB24、エクスポートDB25は、プログラムが更新された時に、更新元の計算機からコピーされた他の計算機の該当プログラムの内容も更新する場合に有効となる。

【0014】中継エージェント2は、ネットワーク3を介して端末装置1や他の計算機（各種サービス4を提供する各計算機）や、他の中継エージェントとの間の通信処理手段により接続が行われ、データ伝送を行うことができる。そして、利用者の端末装置1または他の中継エージェント2からの各種要求に対する処理を行い、ローカルに端末装置1の利用者にサービスを提供したり、他の計算機（ホスト）によりサービスを提供する場合の中継を行う。中継エージェント2の主要な機能を挙げると、次の通りである。

【0015】(1) アクセス方法に関する要求
サービスへアクセスする方法に関する要求であり、アクセス方法DB21に対してアクセス方法の登録要求、削除要求、検索要求を実行する。

【0016】(2) サービスへのアクセス要求
サービスへアクセスするための接続を確立する要求であり、既にサービスが利用可能な状態の時には、そのサービスへの接続を確立し、確立したらアクセス方法DB21を元に、プロトコルの変換を行う等によりサービスへアクセスする。

【0017】サービスが利用可能な状態でない時は、サービスDB22にサービスを提供するプログラムが登録されているか調べて、ある時はそのプログラムを起動しサービスへの接続を確立する。起動するプログラムが、他の計算機に登録されている場合、プログラムのコピーを他の計算機から受け取って、インポートDB24に登録して、起動する。プログラムを渡した計算機では、渡したことを覚えておくためそのエクスポートDB25にどの計算機に渡したかを登録する。

【0018】(3) サービスの登録要求、削除要求、更新要求、検索要求

各要求に含まれるサービスを提供するプログラムの登録、削除、更新及び検索を行う。サービスの登録は、サービスDB22に対して行われ、サービスの削除はサービスDB22だけでなく、関係する現状DB23、インポートDB24、エクスポートDB25の各データベースについても実行される。サービスの更新は、サービスDB22や現状DB23に対して行われる。

【0019】このような構成を備える本発明により、サービスに対するアクセスは、中継エージェント2に対して利用したいサービスへのアクセス要求を送ると、適切なアクセス方法で接続を確立することができる。また、

中継エージェント2がプロトコル変換を行うため、変換する方法が登録されている如何なるプロトコルを用いたサービスに対してでも、適切なアクセス方法で接続を確立することができる。

【0020】また、中継エージェント2に対して利用したい新しいサービスの登録要求を送り、次にそのサービスへのアクセス方法の登録要求を送り、最後にそのサービスへのアクセス要求を送ることにより、適切なアクセス方法で接続を確立することができる。また、中継エージェントは他の計算機上にサービスを提供するプログラムをコピーさせて起動させたり（エクスポートDB25に登録）、他の計算機上からサービスを提供するプログラムをコピーしてきてそれを起動する（インポートDB24に登録してキャッシュの形式で使用する）ことができる。このように必要に応じて必要なサービスを起動することが可能となる。更に、サービスの削除、更新や、各データベースの登録、更新、削除についても適切な要求を中継エージェント2に送ることにより行うことができる。

【0021】

【発明の実施の形態】上記図1に示す中継エージェント2は、制御部20を構成するCPU、メモリを備えると共に各データベース21～25を構成するデータを格納する記憶装置（磁気ディスク装置等）を備え、制御部20は端末装置1の利用者からの要求または他の中継エージェント2からの各種の要求に応じて対応する制御処理を繰り返し実行する。

【0022】最初に、利用者端末または他の中継エージェントからの各種要求の処理において用いられる各データベース（DB）のデータ構造を説明する。図2はアクセス方法DB（図1の21）のデータ構造である。このデータは、先頭にサービス名、次にサービスを提供するサービスホスト、最後にサービスへのアクセス方法の3つのデータで構成され、アクセス方法の例としては、HTTP、AGENT（エージェント用のプロトコル）等のプロトコルがある。図2の下側にアクセス方法DBのデータの例として、（スケジュール管理, bishop, HTTP）とその意味を示す。

【0023】図3はサービスDB（図1の22）及びインポートDB（図1の24）のデータ構造である。これらのDB22、24のデータは、サービス名、プログラム（サービスを提供するプログラム）、プログラムタイプの3つのデータで構成される。プログラムタイプの例としては図3に示すように、C（C言語）、AGENT（モバイルコード：完全なプログラムになってないモジュールで、実行形式であれば良い）、REMOTE（リモートホスト上のプログラム）等がある。これらのDBのデータの例が図3の下側に例1、例2として示す。

【0024】図4は現状DB（図1の23）及びエクス

ポートDB（図1の25）のデータ構造である。これらのDB23、25のデータは、サービス名とサービスホストの2つで構成され、データの具体例を図4の下側に示す。

【0025】上記図2乃至図4に示すデータ構造を備える各データベースを用いて、各中継エージェントにおいて各種要求の処理が行われる。各種の処理内容をそれぞれ処理フローの図として示して説明する。

【0026】サービス方法の登録、削除、更新、検索の各処理フローは図5乃至図8に示す。アクセス方法の登録要求が発生すると、図5に示すアクセス方法の登録要求の処理フローが動作する。入力として“サービス名、サービスを提供するホスト名、アクセス方法”のそれぞれが指定される。これに応じてアクセス方法DB21中に入力されたデータが存在するか判別する（図5のS1）。存在する場合は登録失敗となるが、存在しない場合は、これらの入力されたデータがアクセス方法データ21に登録が行われると共にその登録が成功したか判別し（同S2）、登録成功または、登録失敗の結果が出力される。

【0027】アクセス方法の削除要求が発生すると、図6に示すアクセス方法削除要求の処理フローが動作する。この場合も、入力として上記アクセス方法の登録要求の場合（図5）と同様に、“サービス名、サービスを提供するホスト名、アクセス方法”のそれぞれが指定される。これに応じてアクセス方法DB21中に入力されたデータが存在するか判別する（図6のS1）。存在しない場合はデータが無いことを出力する。存在する場合は、該当するデータの削除を行うと共にその削除が成功したか否かを判別し（図6のS2）、結果に応じて削除成功または削除失敗を出力する。

【0028】アクセス方法の更新要求が発生すると、図7に示すアクセス方法の更新要求の処理フローが動作する。この要求には、入力として“サービス名、元のホスト名、元のアクセス方法、更新後のホスト名、更新後のアクセス方法”が指定され、図7の例では“service, host1, access1, host2, access2”が入力されている。この要求に応じて、『service, host1, access1』をアクセス方法DB21から削除して、その削除が成功したか判別する（図7のS1）。これに失敗するとその出力を行い、削除が成功すると更新後のホスト名と更新後のアクセス方法をアクセス方法DB21に登録して、その処理が成功したか判別する（図7のS2）。成功すると更新成功を出力し、成功しないと更新失敗を出力する。

【0029】アクセス方法の検索要求が発生すると、図8に示すアクセス方法の検索要求の処理フローが動作する。この要求には、入力として“サービス名、ホスト名、アクセス方法”のそれぞれが指定される。この要求に応じて、入力されたデータが存在するか判別し、存在する場合は該当データを出力し、存在しないとデータが

無いことを表す出力を発生する。

【0030】サービスへのアクセス要求

サービスへのアクセス要求には、ホスト上でサービスが利用できないとアクセス失敗として終了する要求（アクセス要求1）と、ホスト上でサービスが利用できなくてもインポートDBやエクスポートDBを検索してアクセスを試みる要求（アクセス要求2）とがある。それぞれの処理フローについて説明する。

【0031】サービスへのアクセス要求1の処理フローを図9に示す。このアクセス要求には、入力として、“サービス名、サービスホスト名、アクセス方法”が指定され、これに対し、アクセス方法データ21中に該当するデータが存在するか判別する（図9のS1）。データが存在しないとアクセス失敗となって終了するが、存在する場合はホスト上で指定されたサービス名のサービスが既に利用できる（起動している）かを判別する（図9のS2）。利用できない場合は、アクセス失敗となるが、利用できる場合は、ホスト上の当該サービスへの接続確立を行い、確立が成功したか判別し（図9のS3）、確立に成功するとアクセス成功を出力し、確立に失敗するとアクセス失敗を出力する。

【0032】サービスへの接続を確立した後は、中継エージェント2はアクセス方法DB21に登録されているアクセス方法を元に、図1に示す端末装置1のGUI1aとサービスとの間でプロトコルの変換を行う。基本的なプロトコル変換は、表を用いたコマンドの書き換えによって可能である。このようなリストを用いて、サービス側（サービスを提供する計算機）が指定しているプロトコルをGUI1a側で指定しているプロトコルに変換する。

【0033】図10はプロトコル変換表の例を示す。この例では、サービス側が、「Display」というコマンドを送ってきた場合には、中継エージェントが「画面上に表示を行うコマンド」に変換して、端末装置1のGUI1a側に送る。逆にGUI1a側が、「他のプロセスにメッセージを送るコマンド」を送って来たら、中継エージェント2が「Send」コマンドに変換して送る。このようにすれば、プロトコルの数だけ同様の表を作ることによりプロトコル変換が可能である。

【0034】サービスへのアクセス要求2の処理フロー（その1）～（その4）を図11～図14に示す。図11に示すようにサービスへのアクセス要求2の処理では、上記図9に示すアクセス要求1の処理フローと同様の入力が指定され、同様のステップS1～S3の処理が行われるが、S2においてホスト上で指定されたサービス名のサービスが既に利用できるか判別して、サービスが利用できないと判別された場合にステップS4に移行する点で相違する。ステップS4では、ホストが‘ローカル’かどうかを判別する（サービスが中継エージェントと同じ計算機上で提供されているかどうかをチェック

する)。ここで、ローカルであると判別された場合はの経路で図13の処理へ移行し、ローカルでない場合はの経路で図14の処理に移行する。

【0035】ローカルの場合、図13において、サービスDB22中に、指定されたサービス名(service)にマッチするデータ(service,code,type)が存在するか判別する(図13のS1)。なお、この場合のデータの「service」はサービス名、「code」はプログラム名、「type」はプログラムタイプ(リモート、モジュール等)を表す。データが存在しないとアクセス失敗となり、存在する場合にはデータのtypeがREMOTE(リモート:他のホストが提供するプログラム)か判別する(図13のS2)。リモートでない場合には、の経路で示す処理(後述する図12)に移行するが、リモートの場合、インポートDB24内に指定されたサービス名(service)で指定されたデータ(service,code',type')が存在するか判別する(図13のS3)。なお、この場合のcode',type'は上記図13のステップS1のcode,typeと異なるプログラム、タイプを表す。

【0036】存在する場合は、ステップS7に移行してサービス(service)としてプログラム(code)を起動して成功したか判別する。存在しない場合は、ホストcode(ステップS1で検索されたデータの中にあるcodeを計算機名として指定している)上のサービスDB22中に、指定されたサービス名(service)のプログラムのデータ(service,code',type')が存在するか判別する(図13のS4)。存在しない場合はアクセス失敗となるが、存在する場合は、当該他の計算機から中継エージェント2のインポートDB24にこのデータのコピーを登録し、登録が成功したか判別する(図13のS5)。登録が成功しないと、アクセス失敗となるが、成功すると、ホストcode上のエクスポートDB25に、このサービス名とホスト名(service,host)の登録を行い、登録が成功したか判別する(図13のS6)。登録が成功しない場合は、アクセス失敗となるが、成功した場合は、そのサービス(サービス名code')を起動し、起動が成功したか判別する(図13のS7)。起動しない場合はアクセス失敗となるが、成功するとサービスへの接続を行って、接続が成功したか判別する(図13のS8)。接続が成功するとアクセスが成功し、そのサービスの提供を受けることが可能となる。

【0037】上記図13のステップS2でプログラムのタイプが「リモート」でないと判別された場合、の経路で図12の処理に移行する。この場合、サービスとして指定された「code」を起動して、起動が成功したか判別し(図12のS1)、成功した場合はサービスへの接続を行って接続が成功したか判別する(図12のS2)。成功した場合はアクセス成功となるが、S1、S2で成功しない場合はアクセス失敗が出力される。

【0038】上記図11のステップS4の判別で、

移行した場合(hostがローカルでない場合)、図14の処理が実行される。最初にサービスDB22を検索し、その中に指定されたサービス名(service)のデータ(service,code,type)が存在するか判別し(図14のS1)、無い場合にはアクセス失敗となるが、存在する場合はそのタイプが「リモート」であるか判別する(同S2)。リモートであればアクセス失敗であるが、リモートでない時は、他のホスト(host)上のインポートDB中にこのプログラムのデータ(service,code,type)のコピーの登録を行い、登録が成功したか判別する(図14のS3)。成功しない場合はアクセス失敗となるが、登録が成功すると、中継エージェント2のエクスポートDB25に(サービス名service,ホスト名host)の登録を行い、登録が成功したか判別する(図14のS4)。成功した場合はホストhost上にサービスserviceとしてプログラムcodeを起動し、起動が成功したか判別する(図14のS5)。起動が成功した場合、サービスserviceへの接続を行って接続が成功したか判別し(同S6)、成功するとアクセス成功の出力を発生する。上記S4、S5及びS6において成功しない場合は、アクセス失敗となる。

【0039】プログラムの起動

サービスを提供するプログラムの起動は、サービスDB22に登録されているプログラムとプログラムタイプを元に行う。例えば、プログラムタイプがCの場合、登録されているプログラムはCで記述されたプログラムであることを意味する。従って、このようなプログラムを他の計算機上で起動するためには、Cコンパイラを用いてそのプログラムをコンパイルして実行可能形式にし、それから起動する。また、エージェント技術の枠組で良く用いられるモバイルコードのような形式の場合には、プログラムタイプがAGENT等となっていて、プログラムとしてコードが登録されていることを意味する。このような場合、そのコードを理解するためのインタプリタを起動し、そのインタプリタにコードを渡すことにより起動する。起動するプログラムが他の計算機上に登録されている場合もある。この場合、プログラムタイプがリモート(REMOTE)となっていて、プログラムとして起動すべきプログラムが登録されているホスト名が指定されている。他の計算機上に登録されているプログラムを起動する場合には、起動すべきプログラムのコピーを他の計算機から貰い、インポートDB24に登録すると同時にそのプログラムを起動する。インポートDB24は、他の計算機に登録されているプログラムのキャッシュとして機能する。プログラムのコピーを渡した計算機では、渡したことを覚えておくためにエクスポートDB25にどのサービスをどの計算機に渡したかを登録する。この情報は、インポートDB24にキャッシュしている計算機に更新情報を知らせるため等に用いる。

【0040】図15はプログラム起動の処理フローであ

る。入力としてサービス名 (service)、プログラム (code)、プログラムのタイプ (type) が与えられて開始すると、タイプが「リモート」 (REMOTE) か判別する (図15のS1)。リモートでない場合は、タイプを元に起動方法を調べ (同S2)、起動方法が判明したかの判別をする (同S3)。判明しないと起動失敗となるが、判明するとその起動方法を元にプログラム (code) を起動し、起動に成功したか判別し (同S4)、成功しない場合は起動失敗となり、成功すると現状DB23に起動したサービス名、ホスト名を登録し、登録に成功したか判別する (同S5)。登録に成功すると起動成功となるが、登録に失敗すると起動失敗となる。

【0041】上記図15のステップS1において、プログラムのタイプがリモートであると判別されると、インポートDB24に指定されたサービス名のプログラム (データ service, code', type') が存在するか判別する (図15のS6)。存在しないと起動失敗となるが、存在するとタイプ=type' を元に起動方法を調べ (同S7)、起動方法が判明したか判別する (同S8)。判明した場合にはその判明した起動方法を元にプログラム (code') を起動し、起動に成功したか判別する (図15のS9)。成功しない場合は起動失敗となるが、成功すると次に現状DB23に、起動したサービス名、ホスト名を登録し、登録が成功したか判別し (同S10)、成功の場合は起動成功となり、そうでないと起動失敗となる。

【0042】上記図15のステップS2及びS7において起動方法を調べる場合、図16の起動方法テーブルの構成例に示すようなテーブルを用いる。すなわち、プログラムタイプを元に起動方法テーブルを調べ、そこに登録されている起動手順に沿ってプログラムを起動すれば良い。

【0043】サービスの登録要求

新しいサービスを提供するプログラムを登録する場合、図17に示すサービスの登録要求の処理フローにより行う。この場合、入力としてサービス名 (service)、そのサービスを提供するプログラムコード (code)、そのプログラムコードのタイプ (type) とを供給すると、サービスDB22中にそのデータが存在しないことが判別されると、そのデータをサービスDB22に登録する処理が行われる。

【0044】サービスの削除要求

サービスの削除要求に対しては、サービスを提供するプログラム (データ) を削除する処理が行われ、図18にサービスの削除要求1の処理フロー、図19、図20にサービスの削除要求2の処理フロー (その1)、 (その2) を示す。なお、削除要求2の処理では、他の利用者が既にアクセスしているサービスの削除も行われる。

【0045】図18のサービスの削除要求1の処理フローでは、入力として削除を要求するサービスを特定する

ための、サービス名、プログラム、プログラムのタイプを表す (service, code, type) が供給され、開始するとサービスDB22中に入力されたサービス名、プログラム及びプログラムタイプのデータが存在するか検索する (図18のS1)。存在しないとデータ無しとなるが、存在すると該当するデータをサービスDB22から削除を行い、削除に成功したか判別する (同S2)。削除できないと削除失敗となるが、成功すると、プログラムのタイプが「リモート」 (REMOTE) か判別する (同S3)。

10 リモートでない場合は、エクスポートDBに上記入力されたプログラムのデータが存在するか判別する (図18のS4)。これは、当該プログラムが他の計算機にコピーされているか調べるもので、存在しないと削除成功となり、存在する場合は、該当するデータをエクスポートDB25から削除して、削除に成功したか判別する (図18のS5)。エクスポートDB25からの削除に成功した場合、そのエクスポートDB25に示されたコピーが行われた宛先のホスト名を持つホスト上のインポートDBから、当該サービス名 (service) のデータを削除して、削除に成功したか判別し (同S6)、成功した場合は削除成功となり、そうでないと削除失敗となる。

20 【0046】上記ステップS3において、削除したいサービスのプログラムのタイプが「リモート」の場合、他の計算機からコピーしたサービスプログラムがインポートDB24にあるか識別するため、インポートDB24に要求されたサービス名のデータが存在するか検索する (図18のS7)。存在しない場合は削除成功となるが、存在すると該当するデータをインポートDB24から削除を行い削除が成功したか判別する (同S8)。成功した場合、そのインポートDB24にコピーを行った元のホストである、ホスト名がcodeである計算機のエクスポートDB25から当該サービス名、ホスト名のデータを削除し、削除に成功したか判別し (同S9)、成功した場合は削除成功となる。

40 【0047】図19、図20のサービスの削除要求2の処理フロー (その1)、 (その2) の場合、入力として上記図18と同様の (service, code, type) が設定される、図19のS1～S9の各処理は上記図18に示す対応する各ステップと同様であるため説明を省略するが、図19のステップS4においてエクスポートDB25にサービス名 (service) のデータ (service, host') が存在しないと判別された時、ステップS7においてインポートDB24にサービス名 (service) のデータが存在しないと判別された時、及びステップS6においてインポートDB24からのデータの削除が成功したと判別された時、更にステップS9において他の計算機 (code) 上のエクスポートDBからサービス名 (service)、ホスト名 (host) のデータの削除が成功したと判別された時の各場合に、この経路で図20の処理に移行する。

50 【0048】この場合、図20に示すように、現状DB

23 を検索して、サービス名 (service) のデータが存在するか判別し (図 20 の S1)、無い場合はデータ無しとなるが、有る場合は該当するデータ (service, host) を現状 DB 23 から削除して、削除に成功したか判別する (図 20 の S2)。成功した場合、ホスト (host) 上で起動されているサービス (service) を中止する処理を行い、中止に成功したか判別する (図 20 の S3)。成功した場合には削除成功の出力が発生するが、成功しないと削除失敗の出力が発生する。このサービスの削除要求 2 により、削除されるプログラムで提供されているサービスに既にアクセスしている利用者がいる場合に現在の接続を中止させることができる。

【0049】サービスの更新要求

サービスを提供しているプログラムを更新する要求であり、更新要求 1 と更新要求 2 とがある。

【0050】図 21 はサービスの更新要求 1 の処理フローであり、入力として (service, code 1, type 1, code 2, type 2) が供給され、code 1, type 1 は更新前のプログラム、プログラムタイプを表し、code 2, type 2 が更新後を表す。サービス DB 22 から更新前のサービス名, code 1, type 1 を削除する処理を行い (図 21 の S1)、削除に失敗すると更新失敗となるが、成功するとサービス DB 22 に code 2, type 2 を登録して、その処理が成功したか判別する (同 S2)。これに成功すると更新成功となる。なお、図 21 では省略されているが、更新の対象となるサービスがエクスポート DB 25 に登録されている場合は、type の計算機がそのサービスを提供するプログラムのコピーを持っているので、そのコピーを持っている計算機 (エクスポート DB 25 の登録内容から認識) 上のインポート DB に登録されている関連情報も更新する。

【0051】図 22 はサービスの更新要求 2 の処理フローであり、更新されるプログラムで提供されているサービスに既にアクセスしている利用者がいる場合、現在の接続を中断させ更新されたプログラムで提供されるサービスに対し接続を再開するようにした。

【0052】入力として供給されるデータは上記図 21 の場合と同様であり、最初にサービス DB 22 中に “service, code 1, type 1” の DB が存在するか検索し (図 22 の S1)、存在する場合は該当するデータをサービス DB 22 から削除する処理を行い、削除に成功したか判別する (同 S2)。成功した場合、現状 DB 23 を検索して、サービス名が service のデータが存在するか検索する (図 22 の S3)。存在する場合は、該当するデータ (service, host) を現状 DB 23 から削除する処理を行って、削除が成功したか判別する (図 22 の S4)。現状 DB 23 からのデータの削除が成功した場合、ホスト名が host であるホスト上で起動されているサービス (service) を中止させ、成功したか判別する (同 S5)。中止が成功すると、更新されたデータ (servic

e, code 2, type 2) をサービス DB 22 に登録する処理を行い、処理が成功したか判別する (同 S6)。成功した場合は、(service, code 2, type 2) としてのプログラムの起動を行って、起動に成功したか判別する (同 S7)。成功すると更新成功が出力され、失敗すると更新失敗が出力される。

【0053】サービスの検索要求

図 23 はサービスの検索要求の処理フローである。サービスを提供するプログラムを検索する場合、入力として (service, code, type) が供給されると、サービス DB 中に入力されたサービス名、プログラム名、プログラムタイプのデータが存在するか検索を行って判別され、存在すると該当データが出力され、存在しないとデータが無いことを表す出力が発生する。

【0054】次に中継エージェント 2 に備えられた各データベースの中の、アクセス方法 DB 21、サービス DB 22 に対する操作は、上記に説明した処理において行われるが、現状 DB 23、インポート DB 24 及びエクスポート DB 25 に対する操作について以下に示し、図 24 ~ 図 26 において現状 DB に関する操作を説明する。

【0055】サービス状況の登録要求

図 24 はサービス状況の登録要求の処理フローである。この場合、入力としてサービス名とサービスを提供するホスト名 (service, host) が供給される。最初に現状 DB 23 中に入力されたサービス名、サービスホストのデータが存在するか検索して判別する (図 24 の S1)。存在すると登録に失敗する (既に登録済) が、存在しない場合は、現状 DB 23 にこのサービス名、サービスホストを登録を行い、登録に成功したか判別し (図 24 の S2)、成功すれば登録成功の出力が発生する。

【0056】サービス状況の削除要求

図 25 はサービス状況の削除要求の処理フローである。あるホスト上で提供されているサービスを終わらせる場合に、終わらせるサービスとサービスを起動していたホストを削除するため、入力として削除したいサービス名とホスト名 (service, host) を供給し、現状 DB 23 を検索し、存在すると入力されたデータを削除し、削除が成功したか判別する。

【0057】サービス状況の検索要求

図 26 はサービス状況の検索要求の処理フローである。現在起動しているサービスとそのサービスを提供しているホストに関する情報を検索する場合に行われ、入力として、サービス名とホスト名 (service, host) を指定すると、現状 DB 23 を検索し、存在する場合は該当データを出力し、存在しないとデータ無しを出力する。

【0058】次に図 27 ~ 図 29 にインポート DB に対する操作を示す。

プログラムのインポート登録要求

図 27 はプログラムのインポート登録要求の処理フロー

であり、サービスを提供するプログラムを他の計算機からコピーしてきた時に、そのサービスに関してどの計算機からプログラムをコピーしてきたのかを記録する場合に実行される。入力として登録したい (service, code, type) を指定し、インポートDBに該当するデータが存在するか検索し、存在しない場合は登録を行って、登録成功か判別する。

【0059】プログラムのインポート削除要求

図28はプログラムのインポート削除要求の処理フローであり、サービスを提供するプログラムが更新される等の理由によりコピーしてきたプログラムが不要となった場合に、それに関する情報を削除する場合に行われる。入力として削除したい (service, code, type) を指定し、インポートDBに該当するデータが存在するか検索し、存在しないとデータが無いことを出力し、存在すると該当するデータをインポートDBから削除する。

【0060】プログラムのインポート検索要求

図29はプログラムのインポート検索要求の処理フローであり、現在、どのようなサービスに関してどのような計算機からプログラムをコピーして来ているのかに関する情報を検索する場合に実行される。入力として検索したい (service, code, type) を指定し、インポートDBに該当するデータが存在するか検索し、存在しないとデータが無いことを出力し、存在すると該当データを出力する。

【0061】次に図30～図32にエクスポートDBに対する操作を示す。

プログラムのエクスポート登録要求

図30はプログラムのエクスポート登録要求の処理フローであり、サービスを提供するプログラムのコピーを他の計算機に渡した時に、そのサービスに関してどの計算機にプログラムのコピーを渡したのかを記録する場合に実行する。入力として、登録したい (service, host) を指定し、エクスポートDB中に該当するデータが存在するか検索し、存在しない場合にエクスポートDBに登録を行い、登録成功か判別する。

【0062】プログラムのエクスポート削除要求

図31はプログラムのエクスポート削除要求の処理フローであり、サービスを提供するプログラムが更新された時等の際にコピーを渡したプログラムを無効にしたい場合に、それに関する情報を削除するために実行される。入力として削除したい (service, host) を指定し、エクスポートDB中に該当するデータが存在するか検索し、存在した場合に該当するデータをエクスポートDBから削除して、削除成功か判別する。

【0063】プログラムのエクスポート検索要求

図32はプログラムのエクスポート検索要求の処理フローであり、現在どのようなサービスに関してどのような計算機にプログラムのコピーを渡したのかに関する情報を検索する。入力として、検索したい (service, host)

を指定し、エクスポートDB中に該当するデータが存在するか検索し、存在する場合に該当データを出力し、存在しないとデータ無しを出力する。

【0064】本発明による中継エージェントを備えたネットワークの環境を個人化することができる。図33は個人化のための構成を示す図である。図中、1は端末装置、2'は中継エージェントを表し、中継エージェント2'は図1に示すネットワーク3と同様のネットワーク(図33では図示省略)に接続される。この構成は、図1に示す中継エージェント2(元の中継エージェントという)の構成に対し利用者チェック部26が追加されている。

【0065】この利用者チェック部26を含む新たな中継エージェント2'に送られる利用者や他の中継エージェント等からの要求は、最初に全て利用者チェック部26に送られ、ここで受けた要求の要求依頼者をチェックする。要求依頼者が中継エージェントの利用者(予め登録した利用者名)と同一であれば、その要求は元の中継エージェントに転送され、同一でなければその要求はエラーとされて受け付けられない。

【0066】利用者は、中継エージェント2'を起動する際に、その名前を中継エージェント2'に登録する。登録された利用者の情報は利用者チェック部26に保持される。また、利用者や他の中継エージェントが中継エージェント2'に送る要求には、必ず要求依頼者の情報を付加する。これにより、中継エージェント2'では、誰からその依頼が送られてきたのか知ることができる。

【0067】図34は要求依頼者チェックの処理フローである。入力の「user」は予め登録された利用者名を表し、(sender, request)は利用者(または他の中継エージェント)からの要求を表し、「sender」には要求依頼者名が格納され、「request」には要求内容が格納される。

【0068】中継エージェント2'では、要求が入力されると、予め登録された利用者(user)と入力された要求の要求依頼者(sender)を比較して、一致が検出されると要求(request)を元の中継エージェントに送り、一致が検出されないと要求依頼者(sender)にエラーを送る。

【0069】本発明による効果的な応用例を説明する。

特定のプログラム言語(例えば、Apri1)で記述されたプログラムが提供されるサービスへのアクセス図1の端末装置1のGUI1aとしてWWWのブラウザを用いてApri1で記述された計算機(エージェント)が提供するサービスへアクセスする場合、GUIがWWWブラウザであるため、中継エージェント2とGUI1a間はHTTPのプロトコルに従わなければならない。また、GUIへの出力はHTMLの形式のテキストで出力しなければならない。さらに、中継エージェント2とサービス間ではApri1言語で提供されるメッセージを用いてサービスにアクセスしなければならない。この

場合、中継エージェントは基本的に以下の処理を繰り返す。

【0070】1. WWWブラウザを通した利用者からの要求をHTTPプロトコルで受け取り、その要求をAprilメッセージに変換しサービス側(Apilプロセス)に送る。

【0071】2. サービス側からAprilメッセージとして送られる結果を受け、HTML形式のテキストに変換し、HTTPプロトコルに従ってGUIに送る。

ソフトウェアの自動メンテナンス

本発明は、ソフトウェアの自動メンテナンスにも応用することができる。上記したように、本発明を用いれば他の計算機に登録されているプログラムのコピーを貰い、それを起動してサービスを利用することができる。

【0072】ある計算機上で、あるソフトウェアを配付する場合に、そのソフトの利用者はサービスをDBに、(サービス名、ソフトが配付されているホスト名、REMOTE)と登録すれば、本発明の構成を用いることにより配付元の計算機より配付ソフトのコピーをインポートDBに自動的に取得して利用することができる。これは、自動的にソフトをダウンロードしていることに相当する。また、配付元でソフトの更新があった場合、本発明によりエクスポートDBを調べてコピーを持っている計算機を調べ、その計算機上のインポートDBを自動的に更新する。これによってソフトウェアの自動メンテナンスが可能である。

【0073】 環境の個人化

本発明による技術を個人単位で行う、すなわち中継エージェントを個人単位で用いることにより、個人でカスタマイズされた環境を構築することが可能である。例えば、中継エージェントを完全に個人化し、他の利用者からのアクセスを禁止してしまえば、サービスDBに個人のプライベートな情報を管理するツールを登録して利用することが可能となる。また、中継エージェントを中心に様々なツールを利用することにより、統合的な個人環境を構築することが可能となる。この場合でも、利用者の中継エージェントは他の利用者からのアクセスを禁止するだけなので、通常通り他の計算機上で提供されているサービスにアクセスすることは可能である。

【0074】

【発明の効果】本発明によれば、ネットワーク上で提供される様々なサービスに対して、簡便で柔軟な方法によりアクセスすることが可能となる。また、他の計算機が保持するプログラムのコピーを取って保管すると共に複写元や、複写先のデータベースに関連するデータを管理するデータベースを備えることによりソフトウェアの自動メンテナンスが可能となる。更に、本発明のアクセス方式を個人単位で用いることにより、個人環境の構築が容易となる。

【図面の簡単な説明】

【図1】本発明の原理構成を示す図である。

【図2】アクセス方法DBのデータ構造を示す図である。

【図3】サービスDB及びインポートDBのデータ構造を示す図である。

【図4】現状DB及びエクスポートDBのデータ構造を示す図である。

【図5】アクセス方法の登録要求の処理フローを示す図である。

10 【図6】アクセス方法削除要求の処理フローを示す図である。

【図7】アクセス方法更新要求の処理フローを示す図である。

【図8】アクセス方法検索要求の処理フローを示す図である。

【図9】サービスへのアクセス要求1の処理フローを示す図である。

【図10】プロトコル変換表の例を示す図である。

20 【図11】サービスへのアクセス要求2の処理フロー(その1)を示す図である。

【図12】サービスへのアクセス要求2の処理フロー(その2)を示す図である。

【図13】サービスへのアクセス要求2の処理フロー(その3)を示す図である。

【図14】サービスへのアクセス要求2の処理フロー(その4)を示す図である。

【図15】プログラム起動の処理フローを示す図である。

【図16】起動方法テーブルの構成例を示す図である。

30 【図17】サービスの登録要求の処理フローを示す図である。

【図18】サービスの削除要求1の処理フローを示す図である。

【図19】サービスの削除要求2の処理フロー(その1)を示す図である。

【図20】サービスの削除要求2の処理フロー(その2)を示す図である。

【図21】サービスの更新要求1の処理フローを示す図である。

40 【図22】サービスの更新要求2の処理フローを示す図である。

【図23】サービスの検索要求の処理フローを示す図である。

【図24】サービス状況の登録要求の処理フローを示す図である。

【図25】サービス状況の削除要求の処理フローを示す図である。

【図26】サービス状況の検索要求の処理フローを示す図である。

50 【図27】プログラムのインポート登録要求の処理フロー

一を示す図である。

【図28】プログラムのインポート削除要求の処理フローを示す図である。

【図29】プログラムのインポート検索要求の処理フローを示す図である。

【図30】プログラムのエクスポート登録要求の処理フローを示す図である。

【図31】プログラムのエクスポート削除要求の処理フローを示す図である。

【図32】プログラムのエクスポート検索要求の処理フローを示す図である。

【図33】個人化のための構成を示す図である。

【図34】要求依頼者チェックの処理フローを示す図で

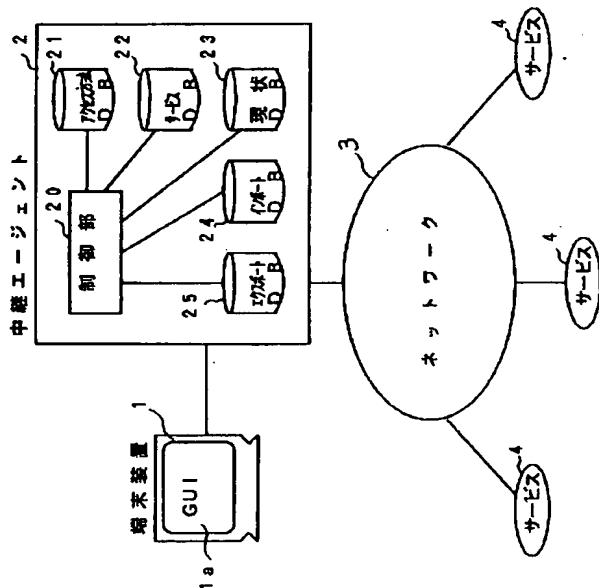
ある。

【符号の説明】

- | | |
|-----|-------------------|
| 1 | 利用者の端末装置 |
| 1 a | G U I |
| 2 | 中継エージェント |
| 2 0 | 制御部 |
| 2 1 | アクセス方法データベース (DB) |
| 2 2 | サービスデータベース (DB) |
| 2 3 | 現状データベース (DB) |
| 2 4 | インポートデータベース (DB) |
| 2 5 | エクスポートデータベース (DB) |
| 3 | ネットワーク |
| 4 | 各種のサービス |

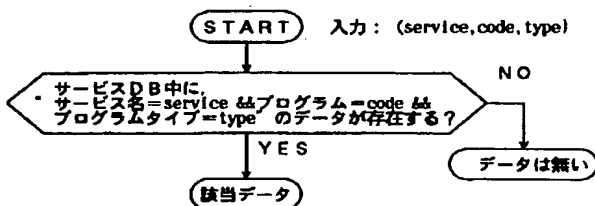
【図1】

本発明の原理構成図



【図23】

サービスの検索要求の処理フロー



【図2】

アクセス方法 (DB) のデータ構造

(サービス名, サービスホスト, アクセス方法)

サービス名: サービスの名前

サービスホスト: サービスが提供されているホスト名

アクセス方法: サービスへのアクセス方法

例: HTTP など... 一般のプロトコル
AGENT ... エージェント用のプロトコル

具体例

(スケジュール管理, bishop, HTTP)

計算機 "bishop" 上で提供されている
サービス "スケジュール管理" に対して
プロトコル "HTTP" を用いてアクセスする

【図3】

サービスDB及びインポートDBのデータ構造

(サービス名, プログラム, プログラムタイプ)

プログラム: サービスを提供するプログラム

プログラムタイプ: プログラムのタイプ

例: C など ... 一般のプログラム
AGENT ... モバイルコード
REMOTE ... リモートホスト上のプログラム

具体例1: (スケジュール管理, CODE_OF_DIARY, C module)

サービス "スケジュール管理" は、
プログラムタイプ "C module" である
プログラム "CODE_OF_DIARY" によって提供される

具体例2: (スケジュール管理, host_name, REMOTE)

サービス "スケジュール管理" は、
プログラムタイプ "REMOTE" であり
ホスト "host_name" 上にサービスを
提供するプログラムが保管されている

【図4】

現状DB及びエクスポートDBのデータ構造

(サービス名, サービスホスト)

サービス名: サービスの名前

サービスホスト: サービスが提供されているホスト名

具体例

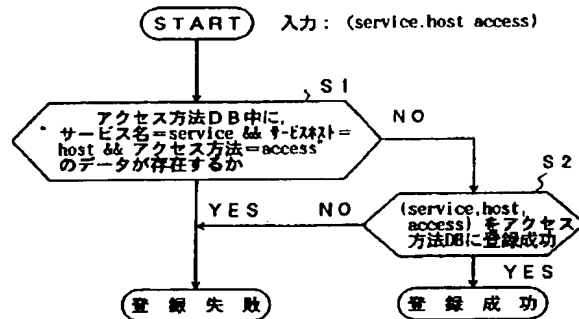
(スケジュール管理, bishop)

↓

計算機 "bishop" 上で, サービス
スケジュール管理 が提供されている

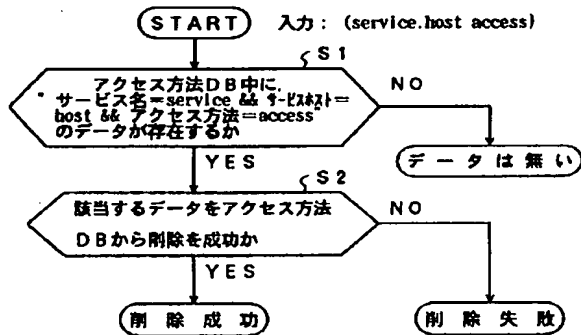
【図5】

アクセス方法の登録要求の処理フロー



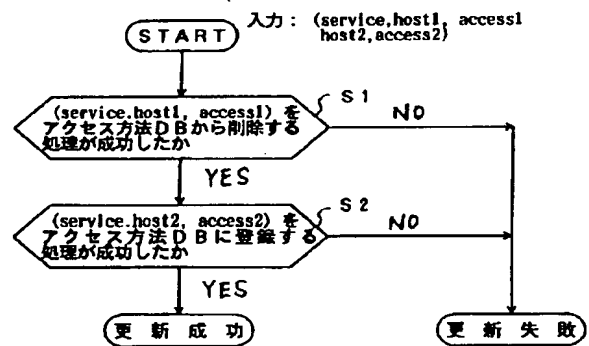
【図6】

アクセス方法の削除要求の処理フロー



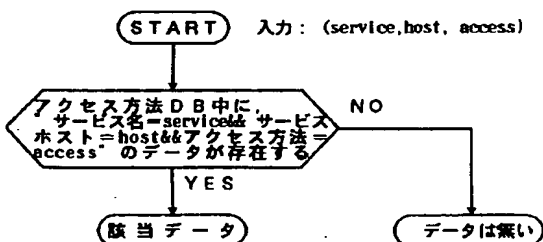
【図7】

アクセス方法の更新要求の処理フロー



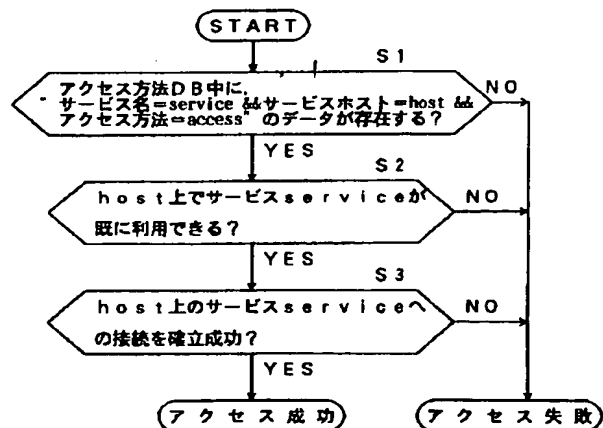
【図8】

アクセス方法の検索要求の処理フロー



【図9】

サービスへのアクセス要求1の処理フロー



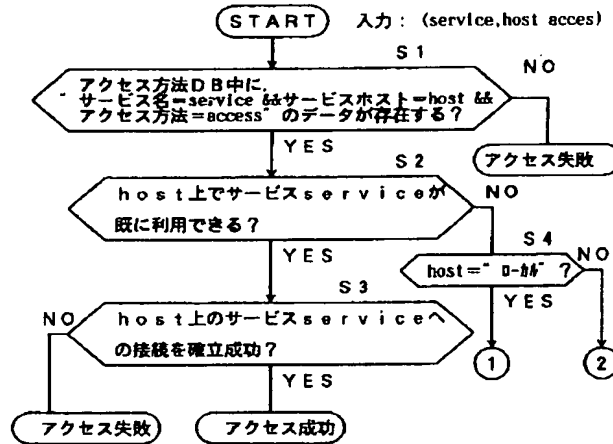
【図10】

プロトコル変換表の例

コマンド名	変換後
Display	画面上に表示を行なう コマンド
Send	他のプロセスにメッセージ を送るコマンド
⋮	⋮

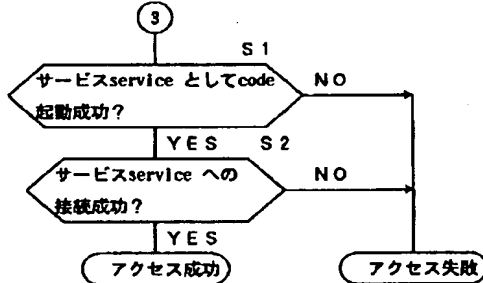
【図11】

サービスへのアクセス要求2の処理フロー（その1）



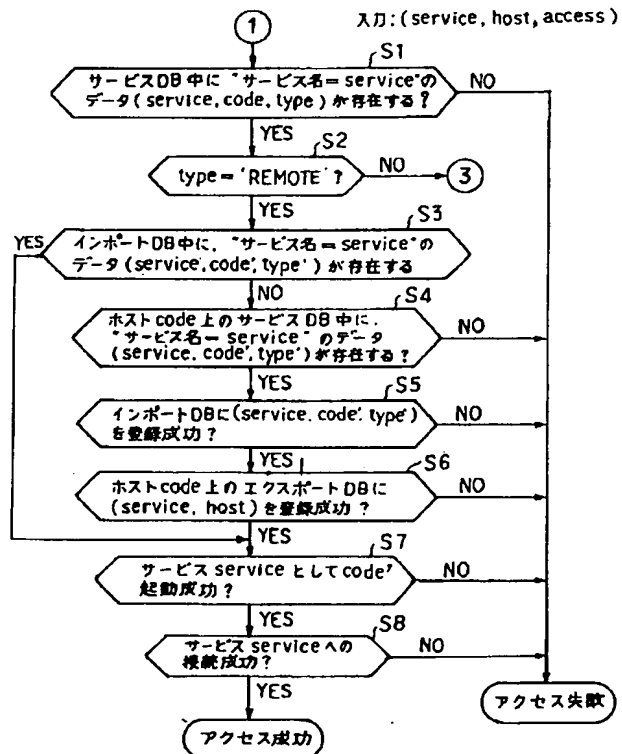
【図12】

サービスへのアクセス要求2の処理フロー（その2）



【図13】

サービスへのアクセス要求2の処理フロー（その3）



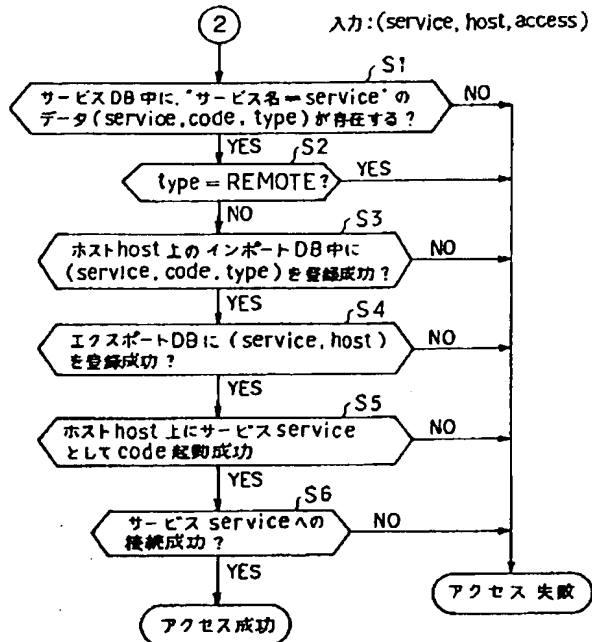
【図16】

起動方法テーブルの構成例

プログラムタイム	起動手順
C	Cコンパイラで実行形式に 変換して起動する
CSH	CSHインタプリタを起動し、 プログラムを解釈させる
⋮	⋮

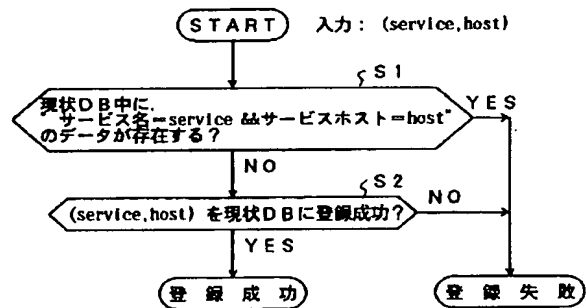
【図 14】

サービスへのアクセス要求2の処理フロー（その4）



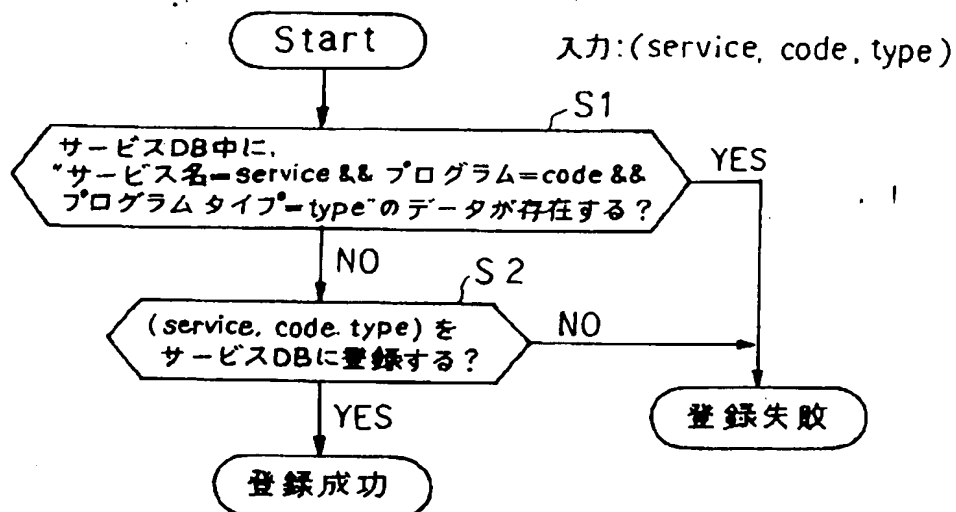
【図 24】

サービス状況の登録要求の処理フロー



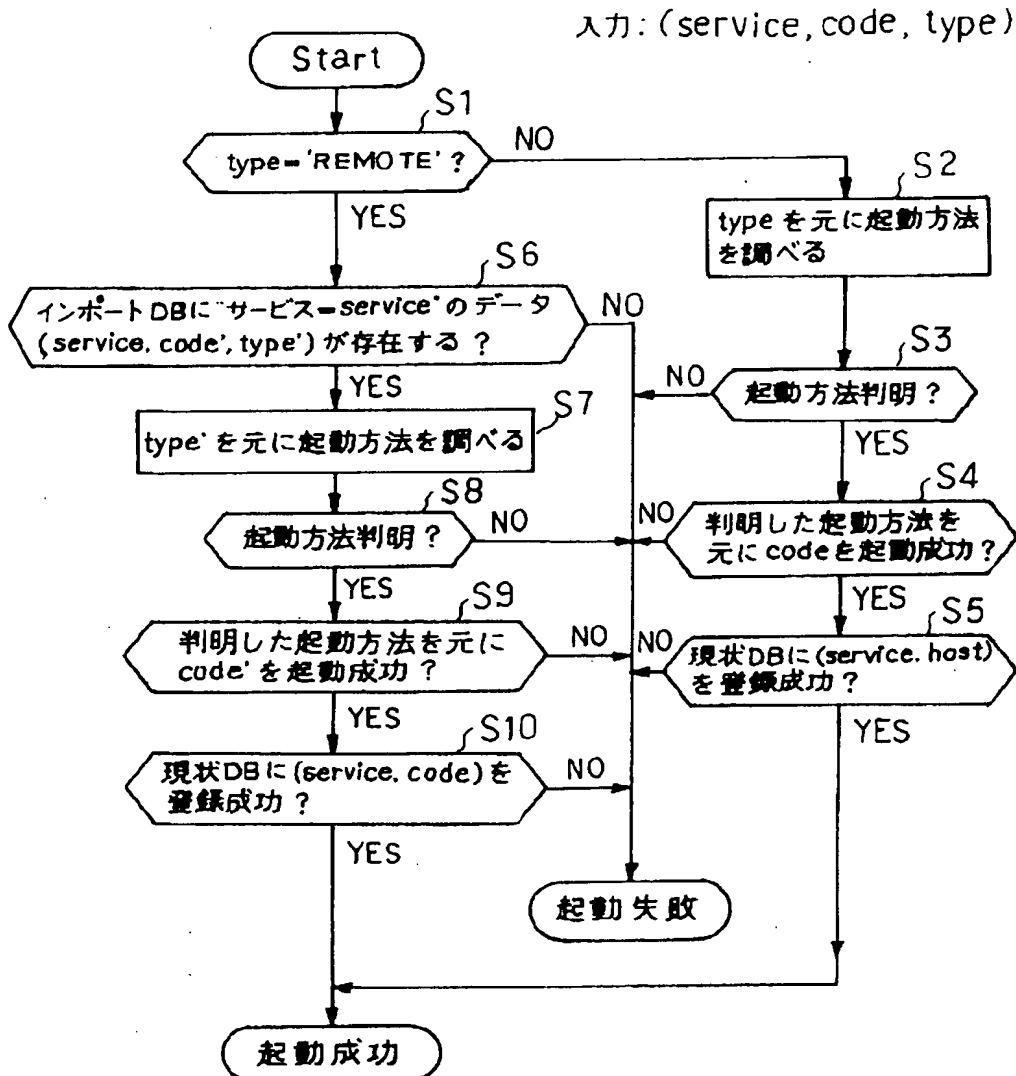
【図 17】

サービスの登録要求の処理フロー



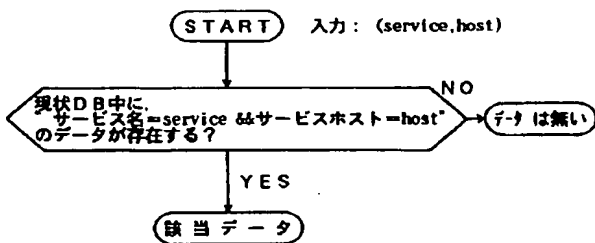
【図15】

プログラム起動の処理フロー



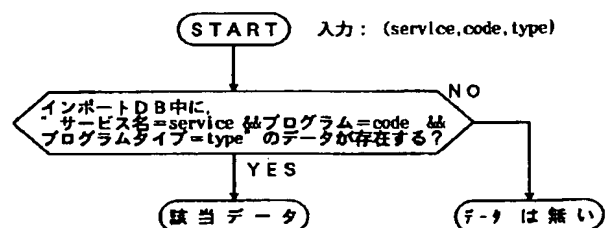
【図26】

サービス状況の検索要求の処理フロー



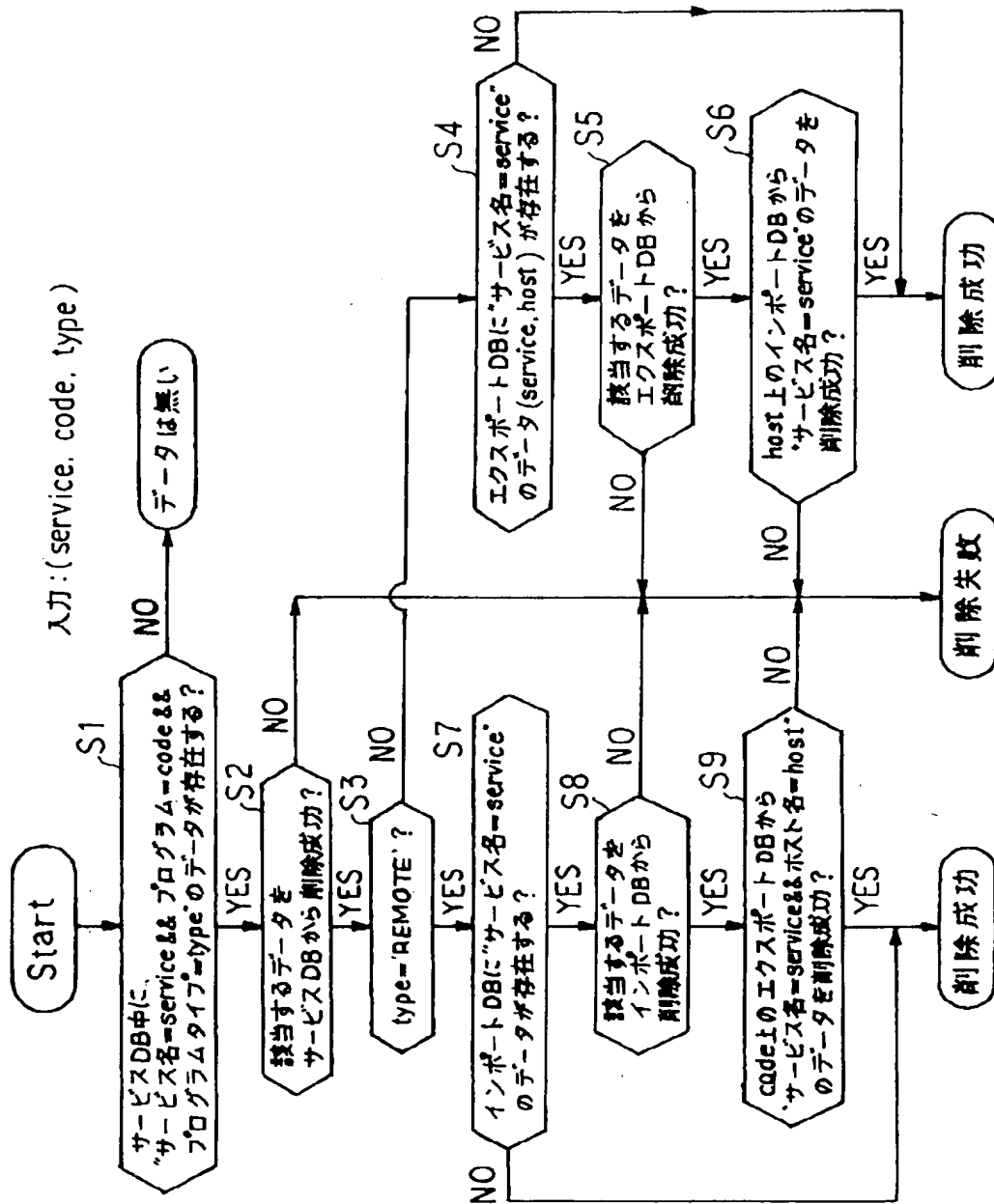
【図29】

プログラムのインポート検索要求の処理フロー



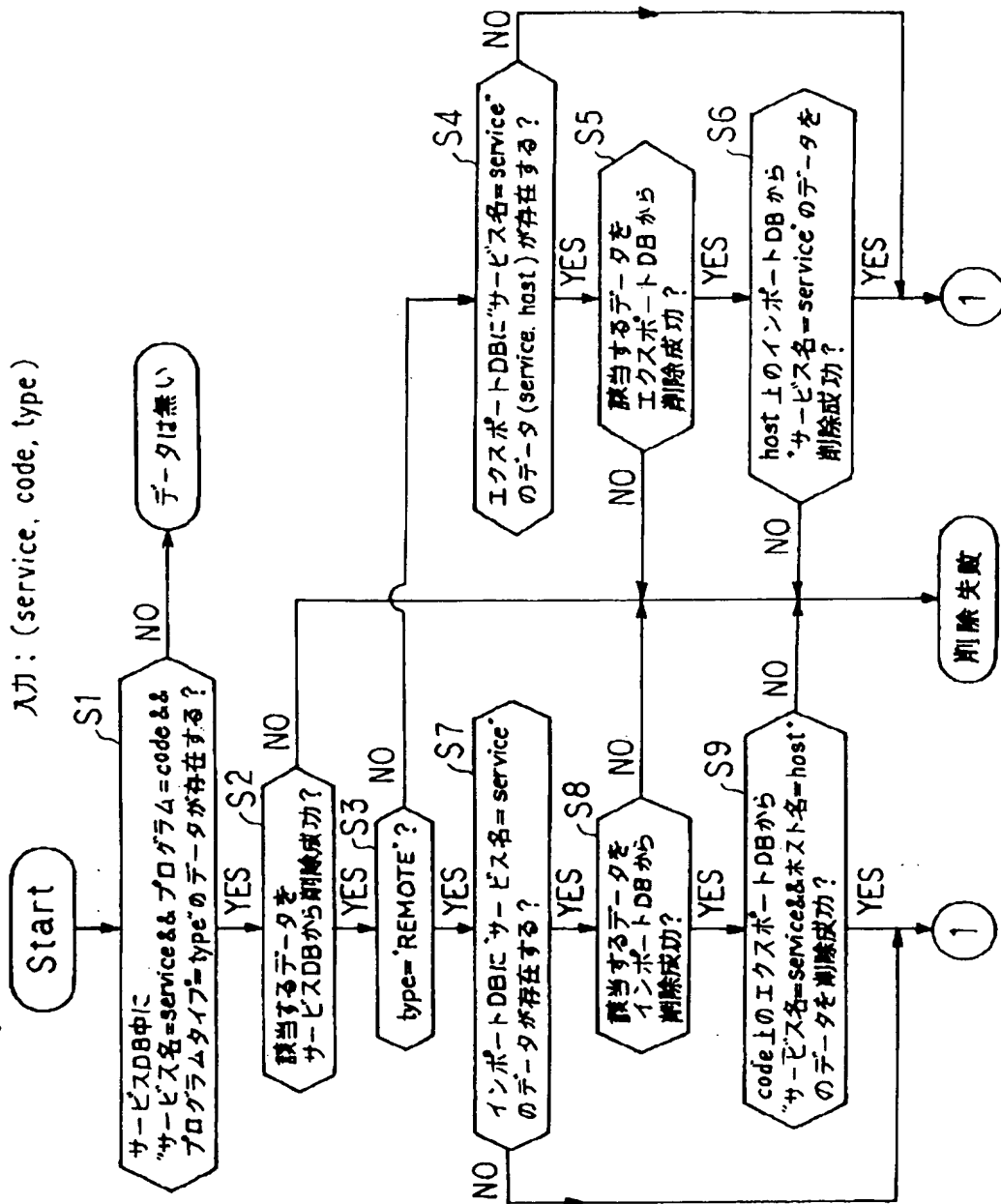
【図18】

サービスの削除要求1の処理フロー



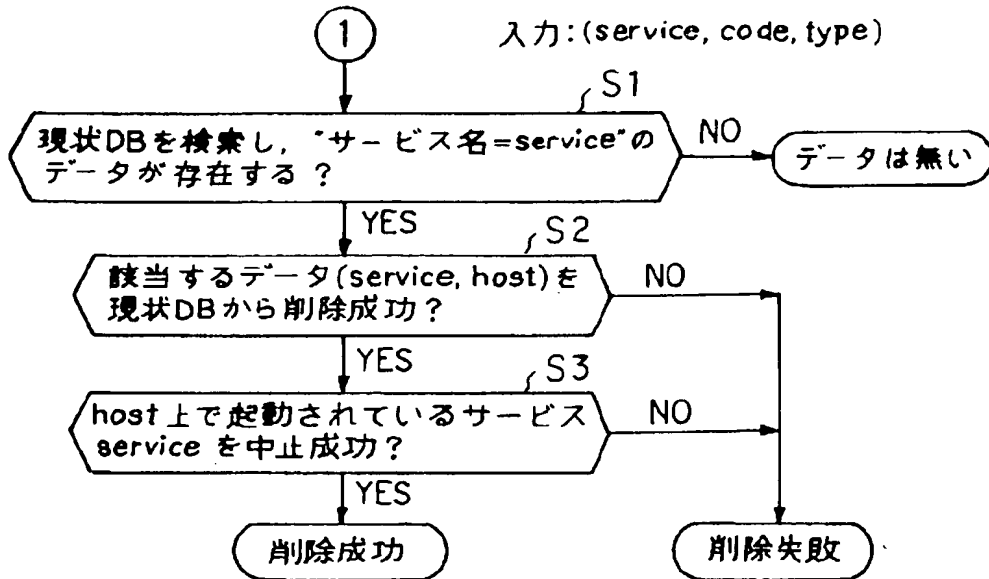
【図19】

サービスの削除要求2の処理フロー（その1）



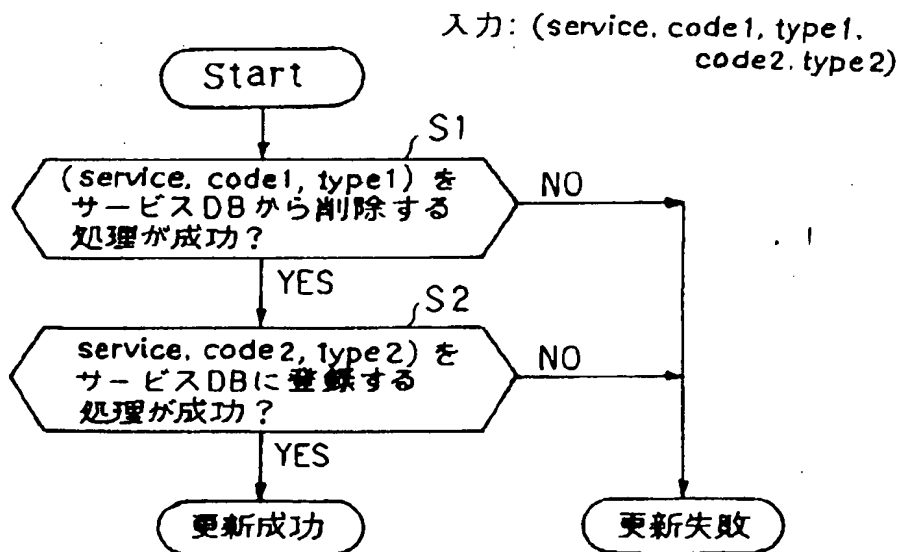
【図20】

サービスの削除要求2の処理フロー（その2）



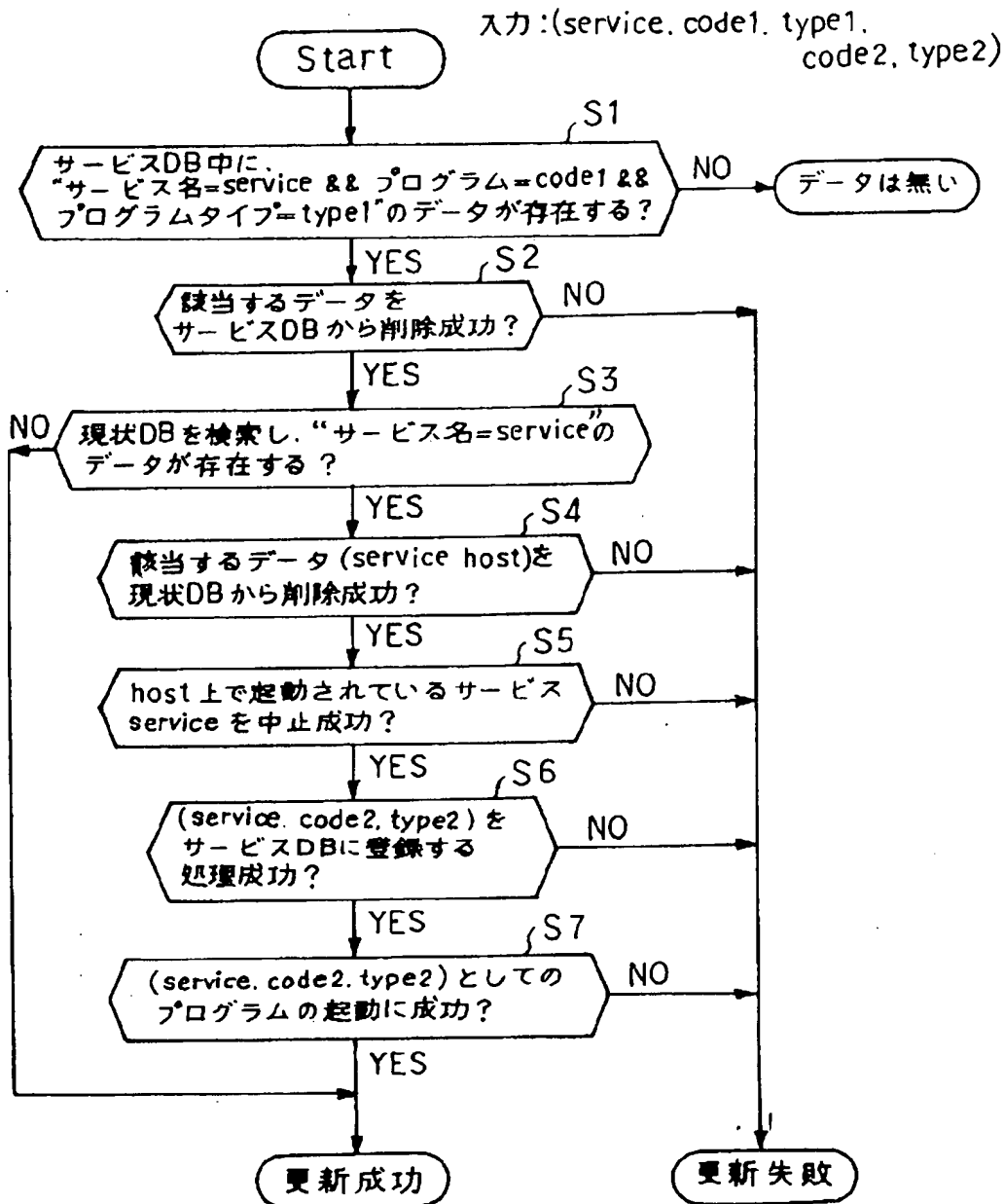
【図21】

サービスの更新要求1の処理フロー



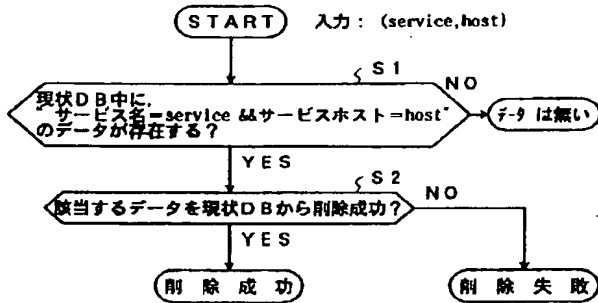
【図22】

サービスの更新要求2の処理フロー



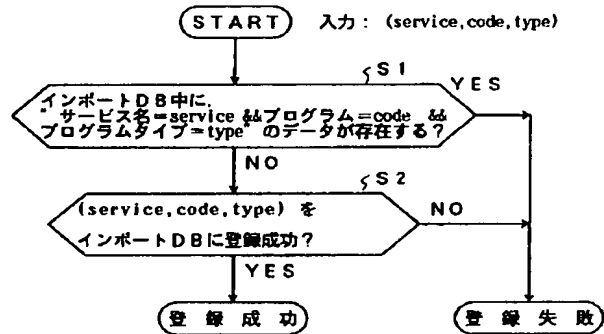
【図25】

サービス状況の削除要求の処理フロー



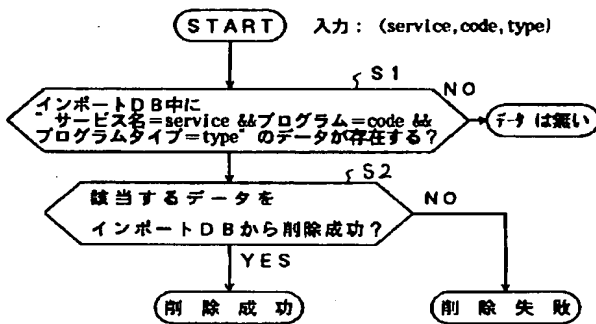
【図27】

プログラムのインポート登録要求の処理フロー



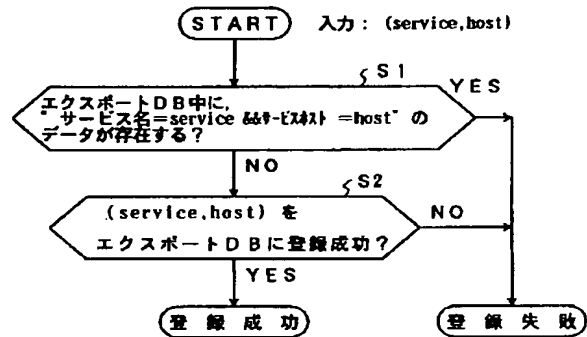
【図28】

プログラムのインポート削除要求の処理フロー



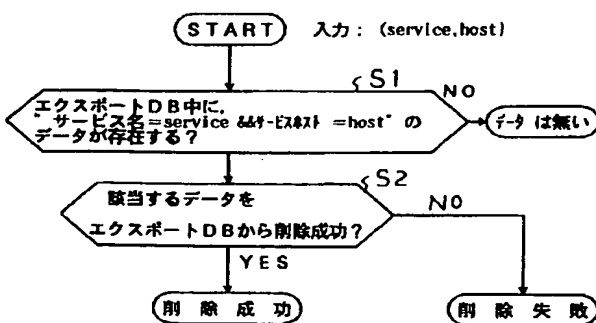
【図30】

プログラムのエクスポート登録要求の処理フロー



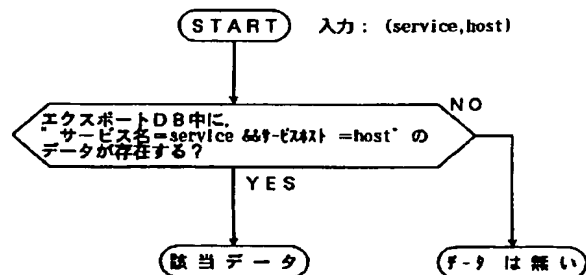
【図31】

プログラムのエクスポート削除要求の処理フロー



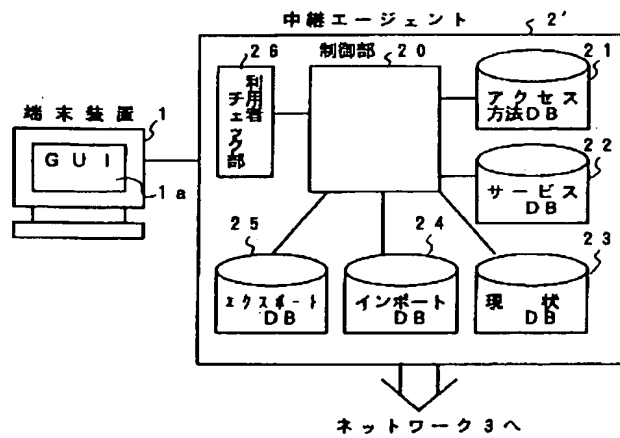
【図32】

プログラムのエクスポート検索要求の処理フロー



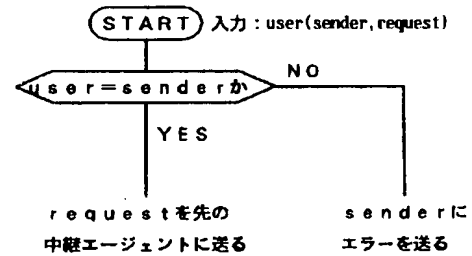
【図33】

個人化のための構成



【図34】

要求依頼者チェックの処理フロー



フロントページの続き

(72)発明者 市来 宏基
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72)発明者 塩内 正利
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72)発明者 高田 裕志
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72)発明者 毛利 隆夫
神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内